ZegCLIP: Towards Adapting CLIP for Zero-shot Semantic Segmentation

Ziqin Zhou[1]   Yinjie Lei[2]   Bowen Zhang[1]   Lingqiao Liu[1*]   Yifan Liu[1]

[1]The University of Adelaide, Australia   [2]Sichuan University, China

{ziqin.zhou,b.zhang,lingqiao.liu,yifan.liu04}@adelaide.edu.au,yinjie@scu.edu.cn

Abstract

*Recently, CLIP has been applied to pixel-level zero-shot learning tasks via a **two-stage** scheme. The general idea is to first generate class-agnostic region proposals and then feed the cropped proposal regions to CLIP to utilize its image-level zero-shot classification capability. While effective, such a scheme requires two image encoders, one for proposal generation and one for CLIP, leading to a complicated pipeline and high computational cost. In this work, we pursue a simpler-and-efficient **one-stage** solution that directly extends CLIP's zero-shot prediction capability from image to pixel level. Our investigation starts with a straightforward extension as our baseline that generates semantic masks by comparing the similarity between text and patch embeddings extracted from CLIP. However, such a paradigm could heavily overfit the seen classes and fail to generalize to unseen classes. To handle this issue, we propose three simple-but-effective designs and figure out that they can significantly retain the inherent zero-shot capacity of CLIP and improve pixel-level generalization ability. Incorporating those modifications leads to an efficient zero-shot semantic segmentation system called **ZegCLIP**. Through extensive experiments on three public benchmarks, ZegCLIP demonstrates superior performance, outperforming the state-of-the-art methods by a large margin under both "inductive" and "transductive" zero-shot settings. In addition, compared with the two-stage method, our one-stage ZegCLIP achieves a speedup of about **5 times faster** during inference. We release the code at* `https://github.com/ZiqinZhou66/ZegCLIP.git`.

1. Introduction

Semantic segmentation is one of the fundamental tasks in the computer vision field, which aims to predict the category of each pixel of an image [7, 15, 31, 41]. Extensive works have been proposed [8, 26, 30], e.g., Fully Convolutional Networks [29], U-net [38], DeepLab family [4–6] and more recently Vision Transformer based methods [16, 53, 58].
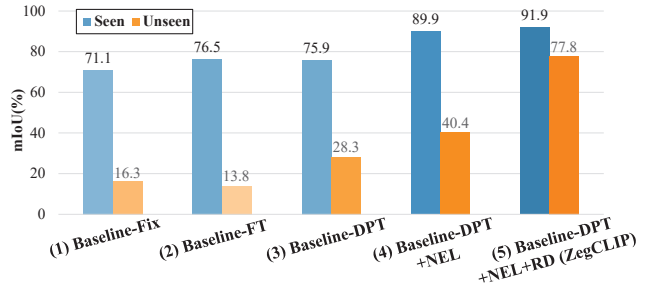

Figure 1. Quantitative improvements achieved by our proposed designs on VOC dataset. (1)(2) represents our one-stage **Baseline** model of different versions (Fix or Fine-Tune CLIP image encoder), while (3)-(5) shows the effectiveness of applying our proposed designs, i.e., **Deep Prompt Tuning (DPT)**, **Non-mutually Exclusive Loss (NEL)**, **Relationship Descriptor (RD)**, on baseline model step by step. We highlight that our designs can dramatically increase the segmentation performance on unseen classes.

However, the success of the deep semantic segmentation models heavily relies on the availability of a large amount of annotated training images, which involves a substantial amount of labor. This gives rise to a surging interest in low-supervision-based semantic segmentation approaches, including semi-supervised [7], weakly-supervised [48], few-shot [46], and zero-shot semantic segmentation [3, 34, 44].

Among them, zero-shot semantic segmentation is particularly challenging and attractive since it is required to directly produce the semantic segmentation results based on the semantic description of a given class. Recently, the pre-trained vision-language model CLIP [36] has been adopted into various dense prediction tasks, such as referring segmentation [43], semantic segmentation [33], and detection [14]. It also offers a new paradigm and has made a breakthrough for zero-shot semantic segmentation. Initially built for matching text and images, CLIP has demonstrated a remarkable capability for image-level zero-shot classification.

However, zsseg [49] and Zegformer [12] follow the common strategy that needs **two-stage** processing that first generates region proposals and then feeds the cropped regions to CLIP for zero-shot classification. Such a strategy requires two image encoding processes, one for generating proposals and one for encoding each proposal via CLIP.

---

*Corresponding author

Table 1. Differences between our approach and related zero-shot semantic segmentation methods based on CLIP.

Methods	Need an extra image encoder?	CLIP as an image -level classifier?	Can do inductive?
zsseg [49]	✓	✓	✓
ZegFormer [12]	✓	✓	✓
MaskCLIP+ [56]	✓	✗	✗
ZegCLIP (Ours)	✗	✗	✓

This design creates additional computational overhead and cannot leverage the knowledge of the CLIP encoder at the proposal generation stage. Besides, MaskCLIP+ [56] utilizes CLIP to generate pseudo labels of novel classes for self-training but will be invalid if the unseen class names in inference are unknown in the training stage ("inductive" zero-shot setting).

This paper pursues simplifying the pipeline by directly extending the zero-shot capability of CLIP from image-level to pixel-level. The basic idea of our method is straightforward: we use a lightweight decoder to match the text prompts against the local embeddings extracted from CLIP, which could be achieved via the self-attention mechanism in a transformer-based structure. We train the vanilla decoder and fix or fine-tune the CLIP image encoder on a dataset containing pixel-level annotations from a limited number of classes, expecting the text-patch matching capability can generalize to unseen classes. Unfortunately, this basic version tends to overfit the training set: while the segmentation results for seen classes generally improve, the model fails to produce reasonable segments on unseen classes. Surprisingly, we discover such an overfitting issue can be dramatically alleviated by incorporating three modified design choices and report the quantitative improvements in Fig. 1. The following highlights our key discoveries:

Design 1: Using **Deep Prompt Tuning (DPT)** instead of fine-tuning or fixing for the CLIP image encode. We find that fine-tuning could lead to overfitting to seen classes while prompt tuning prefers to retain the inherent zero-shot capacity of CLIP.

Design 2: Applying **Non-mutually Exclusive Loss (NEL)** function when performing pixel-level classification but generating the posterior probability of one class independent of the logits of other classes.

Design 3: Most importantly and our major innovation — introducing a **Relationship Descriptor (RD)** to incorporate the image-level prior into text embedding before matching text-patch embeddings from CLIP can significantly prevent the model from overfitting to the seen classes.

By incorporating those three designs into our **one-stage** baseline, we create a simple-but-effective zero-shot semantic segmentation model named **ZegCLIP**. Tab. 1 summarizes the differences between our proposed method and existing approaches based on CLIP. More details can be found in Appendix. We conduct extensive experiments on three public datasets and show that our method outperforms the state-of-the-art methods by a large margin in both the "inductive" and "transductive" settings.

2. Related Works

Pre-trained Vision Language Models [19, 22, 36, 40] which connects the image representation with text embedding have achieved significant performance on various downstream tasks, like image retrieval [28], dense prediction [37], visual referring expression [43], visual question answering [21] and so on. CLIP [36] as one of the most popular vision-language models is pre-trained via contrastive learning on 400 million text-image pairs and shows powerful zero-shot classification ability. In this work, we explore how to adapt CLIP's pre-trained vision-language knowledge from image-level into pixel-level prediction efficiently.

Semantic Segmentation as a fundamental dense prediction task in computer vision requires annotating each pixel of the input image. Previous work follows two principles: 1) Directly considering semantic segmentation as per-pixel classification task [29, 39, 45, 52, 55]; 2) Decoupling the mask generation and the semantic classification [8, 9, 51]. Both methods achieve significant progress with a pre-defined closed set of semantic classes.

Zero-shot Semantic Segmentation remains an important but challenging task due to the inevitable imbalance problem in seen classes. The target model is required to segment unseen classes after training on seen classes with annotated labels. Previous works like SPNet [44], ZS3 [3], CaGNet [17], SIGN [10], Joint [1] and STRICT [34] follow the strategy that improves the generalization ability of semantic mapping from seen classes to unseen classes.

As the popular pre-trained vision-language model CLIP shows the powerful ability of zero-shot classification, it has been also applied in zero-shot semantic segmentation recently. Zegformer [12] and zsseg [49] develop an extensive proposal generator and use CLIP to classify each region and then ensemble the predicting results. Although it remains CLIP's zero-shot ability at the image level, the computational cost increases inevitably due to classifying each proposal. MaskCLIP+ [56] creatively applies CLIP to generate pseudo annotations on novel classes ("transductive") for self-training. It has achieved competitive performance but will be invalid in "inductive" setting where the names of unseen classes in inference are unavailable while training.

Fine-tuning and Prompt Tuning are different methods to update the parameters of pre-trained model when adopt on downstream tasks. Intuitively, fine-tuning [18, 42] is able to achieve significant performance in fully supervised learning tasks, but is challenged by transfer learning. Recently, prompting tuning that freezes the pre-trained model while introducing a set of learnable prompts [2] provides an alternative way. It has achieved satisfied performance in both
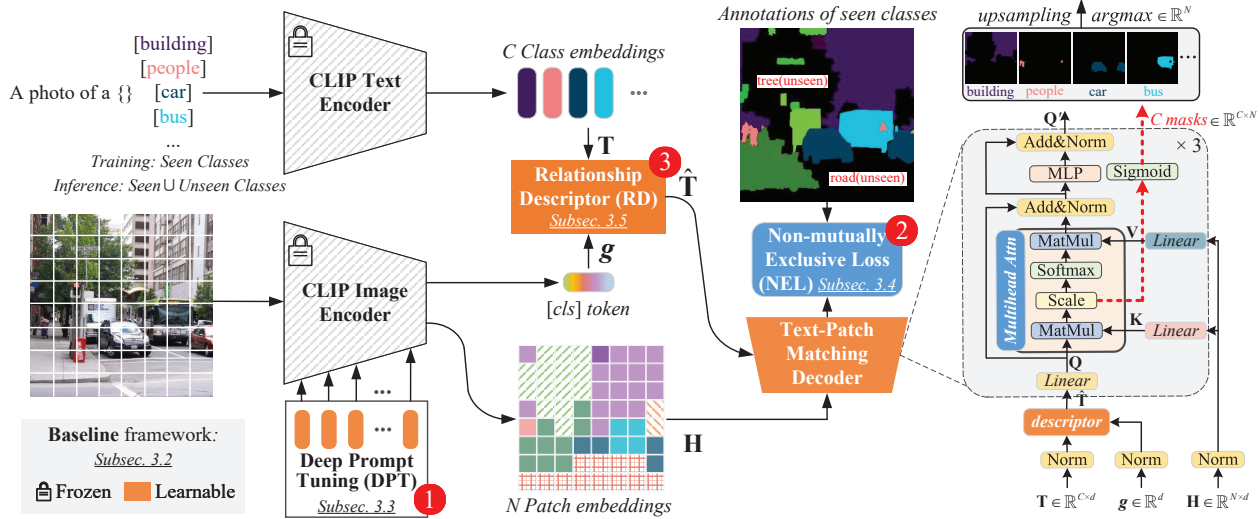
Figure 2. Overall of our proposed **ZegCLIP**. Our method modifies a **One-Stage Baseline** framework of matching text and patch embeddings from CLIP to generate semantic masks. The key contribution of our work is three simple-but-effective designs (labeled as the red circles 1,2,3 in the figure). Incorporating these three designs into our proposed one-stage baseline framework can upgrade the poorly performed baseline method to a strong zero-shot segmentation model.

NLP tasks [24, 27, 35] and CV tasks [47, 50, 57]. Visual Prompt Tuning [20] proposes an effective solution that inserts trainable parameters in each layer of the transformer.

Self-training in Zero-shot Segmentation introduces another setting of zero-shot semantic segmentation called "transductive". Unlike the traditional "inductive" setting where the novel class names and annotations are both unavailable in the training stage, [34] proposed that self-training via pseudo labels on unlabeled pixels benefits solving the imbalance problem. In such "transductive" situation, both the ground truth of seen classes, as well as pseudo labels of unseen classes, will be utilized to supervise the target model for self-training [17, 49, 56] which can achieve better performance.

3. Method

3.1. Problem Definition

Our proposed method follows the generalized zero-shot semantic segmentation (GZLSS) [44], which requires to segment both seen classes $\mathcal{C}^s$ and unseen classes $\mathcal{C}^u$ after only training on a dataset with pixel-annotations of seen part. In the training stage, the model generates per-pixel classification results from the semantic description of all seen classes. In the testing stage, the model is expected to produce segmentation results for both known and novel classes. Note that $\mathcal{C}^s \cap \mathcal{C}^u = \oslash$ and the labels of $\mathcal{C}^u$ are unavailable while training. The key problem of zero-shot segmentation is that merely training on seen classes inevitably leads to overbias on known categories while inference.

This naturally corresponds to the "inductive" zero-shot segmentation setting, in which both unseen class names and images are not accessible during training. Besides "inductive" zero-shot segmentation, there is a "transductive" zero-shot learning setting, which assumes that the names of unseen classes are known before the testing stage. They [17, 56] suppose that the training images include the unseen objects, and only ground truth masks for these regions are not available. Our method can easily be extended to both settings and achieve excellent performance.

3.2. Baseline: One-stage Text-Patch Matching

As the large-scale pre-trained model CLIP shows impressive zero-shot classification ability, recent methods explore applying CLIP to zero-shot segmentation by proposing a **two-stage** paradigm. In stage 1, they train a class-agnostic generator and then leverage CLIP as a zero-shot image-level classifier by matching the similarity between text embeddings and [cls] token of each proposal in stage 2. While effective, such a design requires two image encoding processes and brings expensive computational overhead.

To simplify the two-stage pipeline when adapting CLIP to zero-shot semantic segmentation, in this work, we aim to cope with the critical problem that how to transfer CLIP's powerful generalization capability from image to pixel level classification effectively. Motivated by the observation of recent work [56] that the text embedding can be implicitly matchable to patch-level image embeddings, we build a **one-stage** baseline by adding a vanilla light-weight transformer as a decoder inspired by [13, 51]. Then we formulate semantic segmentation as a matching problem between a representative class query and the image patch features.

Formally, let's denote the C class embeddings as $\mathbf{T} = [\mathbf{t}^1, \mathbf{t}^2, ..., \mathbf{t}^C] \in \mathbb{R}^{C \times d}$, with d is the feature dimension of CLIP model, $\mathbf{t}^i$ representing the ith class, and the N patch tokens of an image as $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N] \in \mathbb{R}^{N \times d}$, with $\mathbf{h}_j$ denoting the jth patch.

Then we apply linear projections ϕ to generate $\mathbf{Q}(query)$, $\mathbf{K}(key)$ and $\mathbf{V}(value)$ as:

$$\mathbf{Q} = \phi_q(\mathbf{T}) \in \mathbb{R}^{C \times d}$$
$$\mathbf{K} = \phi_k(\mathbf{H}) \in \mathbb{R}^{N \times d}, \mathbf{V} = \phi_v(\mathbf{H}) \in \mathbb{R}^{N \times d}. \quad (1)$$

The semantic masks could be calculated by the scaled dot-product attention which is the intermediate product of the multi-head attention model (MHA):

$$\texttt{Masks} = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \in \mathbb{R}^{C \times N}, \quad (2)$$

where $\sqrt{d_k}$ is the dimension of the keys as a scaling factor and the final segmentation results are obtained by applying `Argmax` operation on the class dimension of `Masks`. The detailed architecture of the decoder has been shown on the right of Fig. 2, which consists of three layers of transformers.

How to update the CLIP image encoder: The patch feature representation is generated by the CLIP image encode. How to update the CLIP image encoder, e.g., how to calculate $\mathbf{H}$, is an important factor. In our baseline approaches, we consider $\mathbf{H}$ is obtained from a parameter fixed CLIP or a parameter tunable CLIP, denoted as Baseline-Fix and Baseline-FT separately. Later in SubSec. 3.3, we will discuss Deep Prompt Tuning (DPT), which turns out to be a better way to adapt the CLIP for zero-shot segmentation.

How to train the segmentation model: To properly train the decoder and (optionally) the CLIP model, we apply the commonly used softmax operator to convert the logits calculate from Eq. 2 to the posterior probability. Exclusive Loss (EL) like Cross-entropy is then used as the objective function. Later in SubSec. 3.4, we will point out this seemly straightforward strategy can be potentially harmful for generalization.

Design of query embedding T: The query embeddings $\mathbf{T}$ are the key to our approach. In our baseline model, we use the embeddings from the CLIP text encoder. However, as will be pointed out in SubSec. 3.5, such a choice might cause severe overfitting and we propose a relationship descriptor by using the relationship between text and image token as class queries. In SubSec. 4.5-A, we further explore other choices of $\mathbf{T}$. Please refer to the relevant sections for more details.

As shown in Fig. 1-(1)(2) and Tab. 5, our baseline model turns out to perform poorly in practice, especially for the unseen classes. It seems that the model overfits severely to the seen classes and forgets the zero-shot learning capability of CLIP during training. Fortunately, we identify three
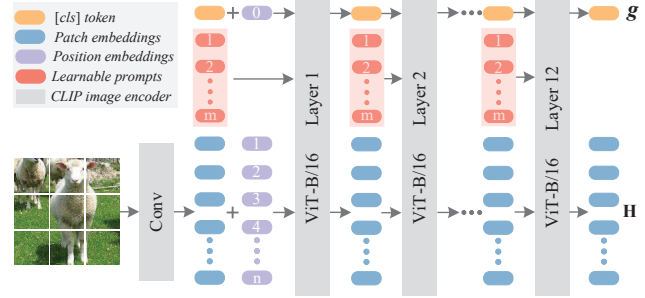

Figure 3. The architecture of deep prompt tuning.

simple-but-effective designs that can dramatically alleviate this issue, as will be described in the subsequent sections.

3.3. Design 1: Deep Prompt Tuning (DPT)

The first design modification is to use deep prompt tuning (DPT) [57] for the CLIP backbone rather than fine-tuning CLIP. As described in Sec. 2, prompt tuning [47] is a recently proposed scheme for adapting a pre-trained transformer model to a target domain. It has become a competitive alternative to fine-tuning in the transfer learning setting [20]. In this work, we explore deep prompt tuning to adapt CLIP for zero-shot image segmentation. Prompt tuning fixes the original parameters of CLIP and adds learnable prompt tokens as additional input for each layer. Since the zero-shot segmentation model is trained on a limited number of seen classes, directly fine-tuning the model tends to overfit the seen classes as the model parameters are adjusted to optimize the loss only for the seen classes. Consequently, knowledge learned for vision concepts unseen from the training set might be discarded in fine-tuning. Prompt tuning could potentially alleviate this issue since the original parameters are intact during training.

Formally, we denote the input embeddings from the l-th MHA module of the ViT-based image encoder in CLIP as $\{\mathbf{g}^l, \mathbf{h}_1^l, \mathbf{h}_2^l, \cdots, \mathbf{h}_N^l\}$, where $\mathbf{g}^l$ denotes the [CLS] token embedding and $\mathbf{H}^l = \{\mathbf{h}_1^l, \mathbf{h}_2^l, \cdots, \mathbf{h}_N^l\}$ denotes the image patch embeddings. Deep prompt tuning appends learnable tokens $\mathbf{P}^l = \{\mathbf{p}_1^l, \mathbf{p}_2^l, \cdots, \mathbf{p}_M^l\}$ to the above token sequence in each ViT layer of CLIP image encoder. Then the l-th MHA module process the input token as:

$$[\mathbf{g}^l, \_, \mathbf{H}^l] = \texttt{Layer}^l([\mathbf{g}^{l-1}, \mathbf{P}^{l-1}, \mathbf{H}^{l-1}]) \quad (3)$$

where the output embeddings of $\{\mathbf{p}_1^l, \cdots, \mathbf{p}_M^l\}$ are discarded (denoted as $\_$) and will not feed into the next layer. Therefore, $\{\mathbf{p}_1^l, \mathbf{p}_2^l, \cdots, \mathbf{p}_M^l\}$ merely acts as a set of learnable parameters to adapt the MHA model.

As shown in Fig. 1-(3) and Tab. 5, compared with fine-tuning, deep prompt tuning on CLIP image encoder can achieve similar performance on seen classes but improve the segmentation results on unseen significantly.

3.4. Design 2: Non-mutually Exclusive Loss (NEL)

The general practice of semantic segmentation models treats it as a per-pixel multi-way classification problem, and Softmax operation is used to calculate the posterior probability, followed by using mutually Exclusive Loss (**EL**) like Cross Entropy as the loss function. However, Softmax essentially assumes a mutually exclusive relationship between the to-be-classified classes: a pixel has to belong to one of the classes of interest. Thus only the relative strength of logits, i.e., the ratio of logits, matters for posterior probability calculation. However, when applying the model to unseen classes, the class space will be different from the training scenario, making the logit of an unseen class poorly calibrated with the other unseen classes.

To handle this issue, we suggest avoiding the mutual-exclusive mechanism during training time and using Non-mutually Exclusive Loss (**NEL**), more specifically, Sigmoid and Binary Cross Entropy (BCE) loss to ensure the segmentation result for each class is independently generated. In addition, we use the focal loss [25] variation of the BCE loss and combine it with an extra dice loss [32, 54] as previous work [9]:

$$\mathcal{L}_{\texttt{focal}} = -\frac{1}{\text{hw}} \sum_{i=1}^{\text{hw}} (1-y_i)^{\gamma} \times \hat{y} \log(y_i) + y_i^{\gamma} \times (1-\hat{y}_i) \log(1-y_i),$$
(4)

$$\mathcal{L}_{\texttt{dice}} = 1 - \frac{2 \sum_{i=1}^{\text{hw}} y_i \hat{y}_i}{\sum_{i=1}^{\text{hw}} y_i^2 + \sum_{i=1}^{\text{hw}} \hat{y}_i^2},$$
(5)

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\texttt{focal}} + \beta \cdot \mathcal{L}_{\texttt{dice}},$$
(6)

where $\gamma = 2$ balances hard and easy samples and $\{\alpha, \beta\}$ are coefficients to combine focal loss and dice loss. As shown in Fig. 1-(4) and Tab. 5, compared with CE, BCE performs better on unseen classes, and the focal loss and dice loss on BCE [9] boost the performance furthermore as demonstrated in SubSec. 4.5-C.

3.5. Design 3: Relationship Descriptor (RD)

In the above design, the class embeddings extracted from the CLIP text encoder will match against the patch embeddings from the CLIP image encoder in the decoder head. While being quite intuitive, we find this design could lead to severe overfitting. We postulate that this is because the matching capability between the text query and image patterns is only trained on the seen-class datasets.

This motivates us to incorporate the matching capability learned from the original CLIP training into the transformer decoder. Specifically, we noticed that CLIP calculates the matching score between a text prompt and an image by

$$\mathbf{t}^{c\top} \mathbf{g} = \sum_j^d t_j^c g_j = \sum_j^d r_j^c,$$
(7)

where $\mathbf{t}^c \in \mathbb{R}^d$ denotes a text embedding for the cth class and t_j is its j-th dimension; $\mathbf{g}$ is the image embedding ([cls] token); $r_j^c = t_j^c g_j$. We postulate that $\mathbf{r}^c = [r_1^c, r_2^c, \cdots, r_d^c]$ characterizes how the image and text, i.e., the text prompt for representing class c, are matched, and call it text-image **Relationship Descriptor (RD)** denoted as $\mathbf{R} \in \mathbb{R}^{C \times d}$ for all C classes in this work. Then we concatenate RDs with the original text embedding $\mathbf{T} \in \mathbb{R}^{C \times d}$ as image-specific text queries $\hat{\mathbf{T}} = \{\hat{\mathbf{t}}^1, \hat{\mathbf{t}}^2, ..., \hat{\mathbf{t}}^C\} \in \mathbb{R}^{C \times 2d}$ for the transformer decoder. Specifically, with this scheme, the input text query of transformer decoder for each class becomes:

$$\hat{\mathbf{t}} = concat[\mathbf{r}, \mathbf{t}] = concat[\mathbf{t} \odot \mathbf{g}, \mathbf{t}],$$
(8)

where $\odot$ is the Hadamard product. Note that, we apply learnable linear projection layers on both $\hat{\mathbf{T}}$ and $\mathbf{H}$ to make them has the same feature dimension.

We compare the effectiveness of applying the relationship description shown in Fig. 1-(5) as well as in Tab. 5. As seen, it can dramatically improve the segmentation results on both seen and unseen categories. In SubSec. 4.5-A, we further discuss the effect of using different combination formats of element-wise operations for text-image matching in the relationship descriptor and text queries $\hat{\mathbf{T}}$.

4. Experiments

4.1. Datasets

To evaluate the effectiveness of our proposed method, we conducted extensive experiments on three public benchmark datasets, including PASCAL VOC 2012, COCO-Stuff 164K, and PASCAL Context. The unseen classes of each dataset are shown in the Appendix according to previous works [12, 17, 49, 56]. The details of the datasets are elaborated as follows:

PASCAL VOC 2012 contains 10,582 augmented images for training and 1,449 for validation. We also ignore the "background" category and use 15 classes as the seen part and 5 classes as the unseen part.

COCO-Stuff 164K is a large-scale dataset that contains 171 categories with 118,287 images for training and 5,000 for testing. The whole dataset is divided into 156 seen classes and 15 unseen classes.

PASCAL Context includes 60 classes with 4,996 for training and 5,104 for testing. The dataset is divided into 50 known classes (including "background") and the rest 10 classes as used as unseen classes in the test set.

4.2. Evaluation Metrics

Following previous works, we measure pixel-wise classification accuracy (pAcc) and the mean of class-wise intersection over union (mIoU) on both seen and unseen classes, denoted as $mIoU(S)$ and $mIoU(U)$, respectively. We also evaluate the harmonic mean IoU ($hIoU$) among seen and unseen classes.

Table 2. Comparison with the state-of-the-art methods on PASCAL VOC 2012, COCO-Stuff 164K, and PASCAL Context datasets. "ST" represents applying self-training via generating pseudo labels on all unlabeled pixels, while "†"+"ST" denotes that pseudo labels are merely annotated on unseen pixels excluding the ignore part.

Methods	PASCAL VOC 2012				COCO-Stuff 164K				PASCAL Context			
	pAcc	mIoU(S)	mIoU(U)	hIoU	pAcc	mIoU(S)	mIoU(U)	hIoU	pAcc	mIoU(S)	mIoU(U)	hIoU
Inductive												
SPNet [44]	-	78.0	15.6	26.1	-	35.2	8.7	14.0	-	-	-	-
ZS3 [3]	-	77.3	17.7	28.7	-	34.7	9.5	15.0	52.8	20.8	12.7	15.8
CaGNet [17]	80.7	78.4	26.6	39.7	56.6	33.5	12.2	18.2	-	24.1	18.5	21.2
SIGN [10]	-	75.4	28.9	41.7	-	32.3	15.5	20.9	-	-	-	-
Joint [1]	-	77.7	32.5	45.9	-	-	-	-	-	33.0	14.9	20.5
ZegFormer [12]	-	86.4	63.6	73.3	-	36.6	33.2	34.8	-	-	-	-
zsseg [49]	90.0	83.5	72.5	77.5	60.3	39.3	36.3	37.8	-	-	-	-
ZegCLIP (Ours)	**94.6**	**91.9**	**77.8**	**84.3**	**62.0**	**40.2**	**41.4**	**40.8**	**76.2**	**46.0**	**54.6**	**49.9**
Transductive												
SPNet+ST [44]	-	77.8	25.8	38.8	-	34.6	26.9	30.3	-	-	-	-
ZS5 [3]	-	78.0	21.2	33.3	-	34.9	10.6	16.2	49.5	27.0	20.7	23.4
CaGNet+ST [17]	81.6	78.6	30.3	43.7	56.8	35.6	13.4	19.5	-	-	-	-
STRICT [34]	-	82.7	35.6	49.8	-	35.3	30.3	34.8	-	-	-	-
zsseg+ST [49]	88.7	79.2	78.1	79.3	63.8	39.6	43.6	41.5	-	-	-	-
ZegCLIP+ST (Ours)	**95.1**	**91.8**	**82.2**	**86.7**	**68.8**	**40.6**	**54.8**	**46.6**	**77.2**	**46.6**	**65.4**	**54.4**
†MaskCLIP+ [56]	-	88.8	86.1	87.4	-	38.1	54.7	45.0	-	44.4	66.7	53.3
†ZegCLIP+ST (Ours)	**96.2**	**92.3**	**89.9**	**91.1**	**69.2**	**40.7**	**59.9**	**48.5**	**77.3**	**46.8**	**68.5**	**55.6**
Fully Supervised												
ZegCLIP (Ours)	**96.3**	**92.4**	**90.9**	**91.6**	**69.9**	**40.7**	**63.2**	**49.6**	**77.5**	**46.5**	**78.7**	**56.9**

4.3. Implementation Details

Our proposed method is implemented based on the open-source toolbox MMSegmentation [11] with PyTorch 1.10.1. All experiments we provided are based on the pre-trained CLIP ViT-B/16 model and conducted on 4 Tesla V100 GPUs, and the batch size is set to 16 with 512x512 as the resolution of images. For "inductive" zero-shot learning, the total training iterations are 20K for PASCAL VOC 2012, 40K for PASCAL Context, and 80K for COCO-Stuff 164K. In the "transductive" setting, we train our ZegCLIP model on seen classes in the first half of training iterations and then apply self-training via generating pseudo labels in the rest of iterations. The optimizer is set to AdamW with the default training schedule in the MMSeg toolbox.

4.4. Comparison with State-of-the-art methods

To demonstrate the effectiveness of our method, the evaluation results compared with previous state-of-the-art methods are reported in Tab. 2. We also provide the fully supervised learning results as the upper bound to show the performance gap between fully-supervised segmentation and zero-shot segmentation results on unseen classes. The qualitative results on COCO-Stuff 164K are shown in Fig. 4 and more visualization results are provided in Appendix.

From Tab. 2, we can see that our proposed method achieves significant performance under the "inductive" setting and outperforms previous works, especially for unseen classes. This clearly demonstrates the superior generalization capability of our method over the previous approaches. In addition, we also find our approach excels in the "transductive" setting, although our method is not specifically designed for that setting. As seen, our model dramatically im-

Table 3. Efficiency comparison with different metrics. All models are evaluated on a single 1080Ti GPU. #Params represents the number of learnable parameters in the whole framework.

Datasets	Methods	#Params(M) ↓	Flops(G) ↓	FPS ↑
VOC	ZegFormer [12]	60.3	1829.3	1.7
	ZegCLIP	**13.8**	**110.4**	**9.0**
COCO	ZegFormer [12]	60.3	1875.1	1.5
	ZegCLIP	**14.6**	**123.9**	**6.7**

proves the performance on unseen classes while maintaining excellent performance on seen part after self-training.

Fig. 4 shows the segmentation results of the Baseline-Fix version and our proposed ZegCLIP on seen and unseen classes. After applying our designs, ZegCLIP shows impressive segmentation ability on both seen and unseen classes and can clearly distinguish similar unseen categories, for example, the unseen "tree", "grass" and "playingfield" categories in (1).

In addition, to demonstrate the efficiency of our proposed method, we compare the number of learnable parameters and inference speed between our one-stage ZegCLIP and typical two-stage method Zegformer [12] in Tab. 3. Our proposed method achieves significant performance while only requiring approximately 14M learnable parameters with 117 Flops(G) on average which is only **23%** and **6%** of [12], respectively. In terms of frames Per Second (FPS), our method can achieve a speedup of about **5 times** faster during inference compared with two-stage method.

4.5. Ablation Study
A. Effect of different formats of text query $\hat{t}$

In this work, we propose an important design called Relationship Descriptor (RD) as described in SubSec. 3.5. In this module, we combine the queried text embeddings **T**

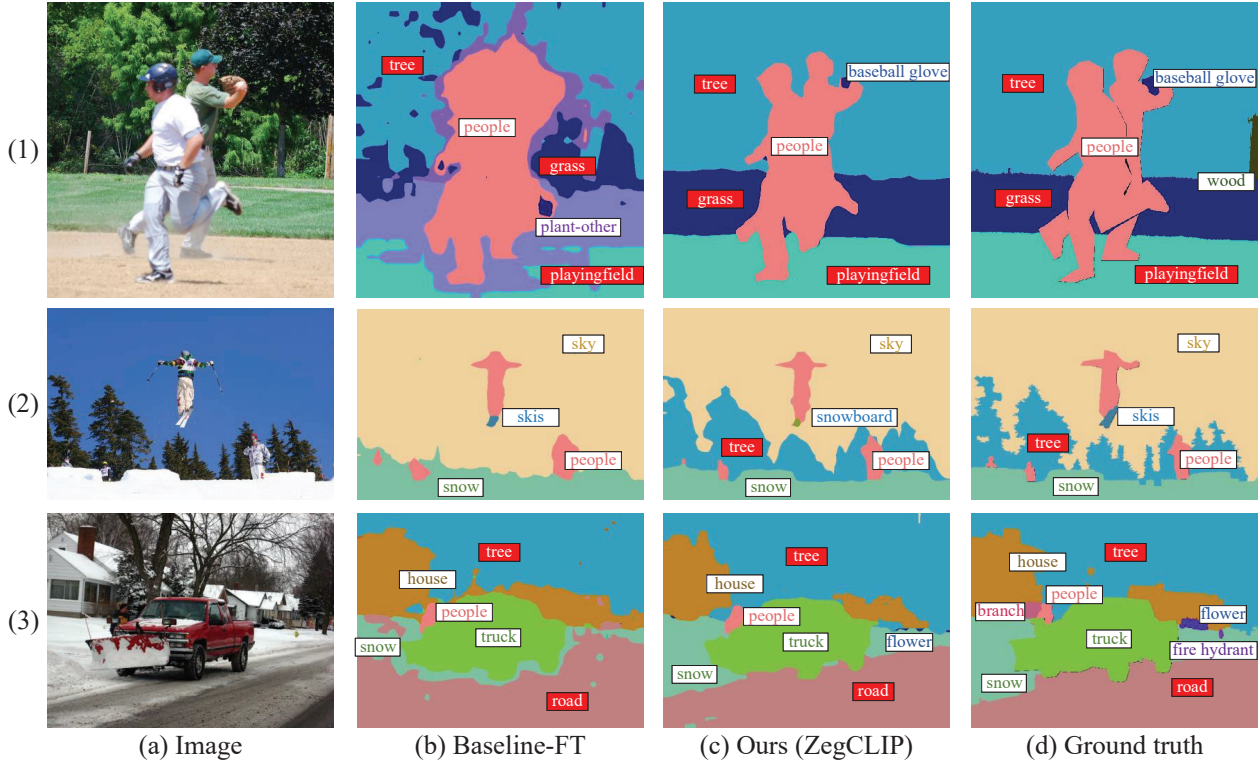| (1) | | | |
(2)			
(3)			
(a) Image	(b) Baseline-FT	(c) Ours (ZegCLIP)	(d) Ground truth

Figure 4. Qualitative results on COCO-Stuff 164K. (a) are the original testing images; (b) represent the performance of our proposed one-stage baseline (fine-tuning the image encoder); (c) are the visualization results of our proposed ZegCLIP; (d) are the ground truths of each image. Note that the white and red tags represent seen and unseen classes separately.

with image [cls] token g extract from CLIP to generate text-image description $\hat{T}$ before feeding the queries into the segment decoder. Such image-specific queries can dramatically improve the zero-shot segmentation performance on unseen classes. To further explore the effect of different element-wise operations in the relationship descriptor, we conduct various formats of $\hat{T}$ on VOC and report the results in Tab. 4. For the image-specific text query of each class $\hat{t}$, it can be formulated as shown in the second column.

We can see, the dot product and the absolute difference between text embedding t and [cls] token g can provide more general information, but the sum and concatenate operations perform poorly on both seen and unseen classes. This is understandable since dot product and absolute difference characterize the relationship between the image and text encoding, which is in line with the interpretation made in SubSec. 3.5. The red line in Tab. 4 is the format of $\hat{t}$ we finally chose in our ZegCLIP model.

B. Detailed results of applying designs on baseline

To demonstrate the effectiveness of our proposed designs, We further report the improvements of applying designs on our Baseline model in Tab. 5. It shows that fixed CLIP (Baseline-Fix) with a learnable segment decoder fails to achieve satisfactory performance on both seen and unseen classes due to weak image representation for the dense

Table 4. Effect of different formats of text queries $\hat{t}$.

dim	format of $\hat{t}$	pAcc	mIoU(S)	mIoU(U)	hIoU
	t	86.8	89.5	33.7	49.0
	$t \odot g$	93.1	90.2	68.4	77.8
512	$|t-g|$	92.4	90.6	64.2	75.1
	$t-g$	88.7	87.9	46.5	60.8
	$t+g$	82.2	89.9	13.9	24.1
	$[t, g]$	88.9	88.8	39.3	54.5
	$[t \odot g, t]$	**94.6**	**91.9**	**77.8**	**84.3**
512*2	$[|t-g|, t]$	90.9	91.5	54.2	68.1
	$[t \odot g, t+g]$	88.3	90.0	38.0	53.4
	$[t+g, t]$	82.8	89.4	20.7	33.6
	$[t \odot g, |t-g|]$	94.1	91.2	73.9	81.6
512*3	$[t \odot g, |t-g|, t]$	93.4	91.6	67.3	77.6

prediction task. Meanwhile, fine-tuning CLIP (Baseline-FT) performs better on seen classes but deteriorates on unseen classes dramatically. It seems that the zero-shot transfer learning ability of CLIP is destroyed after updating on seen classes. Changing objective functions from mutual exclusive into **Non-mutually Exclusive Loss (NEL, Design 2)** also leads to significant performance boost both in seen and unseen scenarios. The most dramatic improvement comes from **Relationship Descriptor (RD, Design 3)** which almost doubles the unseen performance in many cases. Finally, we noticed that **Deep Prompt Tuning (DPT, Design 1)** works well with NEL (Design 2) and RD (Design 3). If we replace DPT by fine-tuning CLIP and combine it with NEL and RD, the issue of overfitting seen classes

Table 5. Quantitative results on VOC and COCO dataset to demonstrate the effectiveness of our proposed three designs.

| method	PASCAL VOC 2012				COCO-Stuff 164K			
	pAcc	mIoU(S)	mIoU(U)	hIoU	pAcc	mIoU(S)	mIoU(U)	hIoU
Baseline-Fix	69.3	71.1	16.3	26.5	33.3	17.1	15.4	16.2
Baseline-Fix + NEL	85.5	85.2	36.6	51.2	52.4	31.7	20.8	25.1
Baseline-Fix + RD	86.0	82.5	46.6	59.6	41.0	23.3	23.4	23.3
Baseline-Fix + NEL + RD	89.6	83.3	66.4	73.9	53.7	32.3	32.5	32.4
Baseline-FT	77.3	76.5	13.8	23.4	48.4	32.4	17.5	22.7
Baseline-FT + NEL	83.8	84.1	27.5	41.4	56.5	39.9	25.4	31.0
Baseline-FT + RD	79.4	77.8	20.7	32.7	54.0	39.6	22.4	28.6
Baseline-FT + NEL + RD	89.6	90.2	42.4	57.7	60.2	**42.7**	22.3	29.3
Baseline-DPT	76.2	75.9	28.3	41.2	39.0	22.5	17.5	19.7
Baseline-DPT + NEL	89.2	89.9	40.4	55.7	58.5	38.0	27.4	31.8
Baseline-DPT + RD	85.5	81.0	55.2	65.7	46.4	28.4	27.8	28.1
Baseline-DPT + NEL + RD (ZegCLIP)	**94.6**	**91.9**	**77.8**	**84.3**	**62.0**	40.2	**41.4**	**40.8**
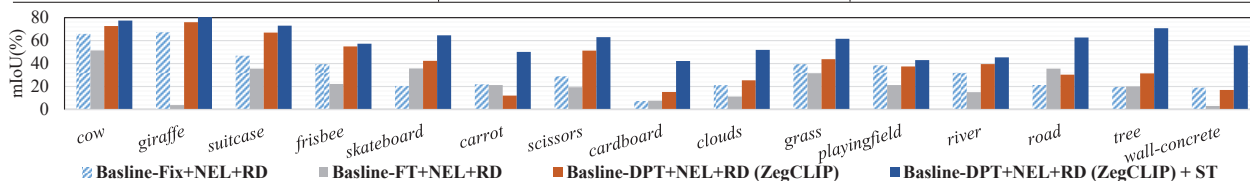
Figure 5. Detailed performance on unseen classes of COCO datasets. Note that "ST" represents self-training in "transductive" setting.

remain. In conclusion, our proposed method can ensure semantic segmentation performance while maintaining the powerful ability of CLIP and achieving excellent performance on seen and unseen classes simultaneously.

C. Effect of advanced loss function

As we described in Subsec 3.4, we propose that applying non-mutually exclusive objective functions like using binary cross entropy (BCE) with sigmoid performs better on zero-shot semantic segmentation. To better handle the imbalance problem among categories, we further combine BCE with focal loss [25] and dice loss [23] according to [9, 51] as Eq. 6, and the results of applying such two versions of the loss function, denoted as "plain" and "plus", are reported separately in Tab. 6. We can see that the "plus" loss function achieves better performance on both seen and unseen classes on VOC, COCO, and Context datasets.

Table 6. Comparison of introducing advanced loss function. Note that "plain" represents merely Binary Cross Entropy (BCE), while "plus" means adding focal loss on BCE and dice loss

dataset	loss	pAcc	mIoU(S)	mIoU(U)	hIoU
VOC	plain	93.4	89.7	73.6	80.9
	plus	**94.6**	**91.9**	**77.8**	**84.3**
COCO	plain	59.8	38.8	39.0	38.9
	plus	**62.0**	**40.2**	**41.4**	**40.8**
Context	plain	75.3	43.5	50.0	46.5
	plus	**76.2**	**46.0**	**54.6**	**49.9**

D. Generalization ability to other datasets

To further explore the generalization ability of our proposed method, we conduct extra experiments in Tab. 7. We apply the pre-trained model of the source dataset via supervised learning on seen classes and evaluate the segmentation results on both seen and unseen classes of target datasets. Our method shows better cross-domain gen-

eralization capability compared with the latest related work Zegformer which is also based on the CLIP model.

Table 7. Generalization ability to other datasets.

source	target	method	pAcc	mIoU	mAcc
COCO	Context	Zegformer [12]	56.8	36.1	64.0
		ZegCLIP	60.9	41.2	68.4
		†ZegCLIP+ST	**68.4**	**45.8**	**70.9**
	VOC	Zegformer [12]	92.8	85.6	92.7
		ZegCLIP	96.9	93.6	96.4
		†ZegCLIP+ST	**97.2**	**94.1**	**96.7**

5. Conclusion

In this work, we propose an efficient one-stage straightforward zero-shot semantic segmentation method based on the pre-trained vision-language CLIP. To transfer the image-wise classification ability to dense prediction tasks while maintaining the advanced zero-shot knowledge, we figure out three designs to achieve competitive results on seen classes while extremely improving the performance on novel classes. Our proposed method relay on text embeddings as queries that are very flexible to cope with both "inductive" and "transductive" zero-shot settings. To demonstrate the effectiveness of our method, we conduct extensive performance on three public benchmark datasets and outperform previous state-of-the-art methods. Meanwhile, our one-stage framework shows about 5 times faster compared with two-stage methods while inference. In general, our work explores how to leverage the pre-trained vision-language model CLIP into semantic segmentation and successfully utilize its zero-shot knowledge in downstream tasks which may provide inspiration for future research.

References

[1] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9536–9545, 2021. 2, 6

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020. 2

[3] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 1, 2, 6

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 1

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017. 1

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[7] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2613–2622, 2021. 1

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, 2022. 1, 2

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 34:17864–17875, 2021. 2, 5, 8

[10] Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed. SIGN: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9556–9566, 2021. 2, 6

[11] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6

[12] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11583–11592, 2022. 1, 2, 5, 6, 8

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[14] Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu. Zero-shot out-of-distribution detection based on the pretrained model clip. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2022. 1

[15] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9716–9725, 2021. 1

[16] Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z Pan. Multi-scale high-resolution vision transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12094–12103, 2022. 1

[17] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, pages 1921–1929, 2020. 2, 3, 5, 6

[18] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018. 2

[19] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4904–4916, 2021. 2

[20] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 3, 4

[21] Jingjing Jiang, Ziyi Liu, and Nanning Zheng. Finetuning pretrained vision-language models with correlation information bottleneck for robust visual question answering. *arXiv preprint arXiv:2209.06954*, 2022. 2

[22] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2

[23] Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*, 2019. 8

[24] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 5, 8

[26] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 82–92, 2019. 1

[27] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 61–68, 2022. 3

[28] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2125–2134, 2021. 2

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1, 2

[30] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *Proceedings of the IEEE conference on European Conference on Computer Vision (ECCV)*, pages 661–679. Springer, 2020. 1

[31] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8364–8375, 2022. 1

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016. 5

[33] Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E Green, and Nassir Navab. Segmentation in style: Unsupervised semantic image segmentation with stylegan and clip. *arXiv preprint arXiv:2107.12518*, 2021. 1

[34] Giuseppe Pastore, Fabio Cermelli, Yongqin Xian, Massimiliano Mancini, Zeynep Akata, and Barbara Caputo. A closer look at self-training for zero-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2693–2702, 2021. 1, 2, 3, 6

[35] Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021. 3

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language super-

vision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 1, 2

[37] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18082–18091, 2022. 2

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 234–241, 2015. 1

[39] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7262–7272, 2021. 2

[40] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[41] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10052–10062, 2021. 1

[42] Deze Wang, Zhouyang Jia, Shanshan Li, Yue Yu, Yun Xiong, Wei Dong, and Xiangke Liao. Bridging pre-trained models and downstream tasks for source code understanding. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 287–298, 2022. 2

[43] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11686–11695, 2022. 1, 2

[44] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8256–8265, 2019. 1, 2, 3, 6

[45] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 34:12077–12090, 2021. 2

[46] Guo-Sen Xie, Jie Liu, Huan Xiong, and Ling Shao. Scale-aware graph neural network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5475–5484, 2021. 1

[47] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. Class-aware visual prompt tuning for vision-language pre-trained model. *arXiv preprint arXiv:2208.08340*, 2022. 3, 4

[48] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6984–6993, 2021. 1

[49] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. *arXiv preprint arXiv:2112.14757*, 2021. 1, 2, 3, 5, 6

[50] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. CPT: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021. 3

[51] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, and Yifan Liu. Segvit: Semantic segmentation with plain vision transformers. *arXiv preprint arXiv:2210.05844*, 2022. 2, 3, 8

[52] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7151–7160, 2018. 2

[53] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12083–12093, 2022. 1

[54] Rongjian Zhao, Buyue Qian, Xianli Zhang, Yang Li, Rong Wei, Yang Liu, and Yinggang Pan. Rethinking dice loss for medical image segmentation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 851–860. IEEE, 2020. 5

[55] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6881–6890, 2021. 2

[56] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *Proceedings of the IEEE conference on European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 5, 6

[57] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 130(9):2337–2348, 2022. 3, 4

[58] Fangrui Zhu, Yi Zhu, Li Zhang, Chongruo Wu, Yanwei Fu, and Mu Li. A unified efficient pyramid transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2667–2677, 2021. 1